

Treball de Fi de Grau

Grau en Enginyeria de Tecnologies Industrials

Disseny i programació d'un sistema de pantalla amb connexió a internet per visualitzar les reserves d'una sala i gestionar el seu horari de disponibilitat

MEMÒRIA

Autor: Gerard Bals Martínez
Director: Oriol Boix Aragonès
Convocatòria: Gener 2020



**Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona**



Resum

En aquest treball s'ha desenvolupat un sistema per tractar la informació guardada en un calendari i mostrar-la de forma visual en pantalles. Aquestes es trobaran a l'exterior dels espais on tenen lloc els esdeveniments programats en aquesta agenda.

En el projecte es realitza la programació d'un microcontrolador que gestiona les dades i es mostra la informació més rellevant del calendari en les pantalles perquè pugui ser llegida de forma ràpida i entenedora.

L'objectiu principal és permetre saber a un usuari, que necessita utilitzar una sala de forma imprevista, quines reserves hi ha a cada sala per poder decidir quina pot ocupar sense afectar a la resta d'usuaris que hagin reservat aquest espai de forma prèvia. També s'intenta aconseguir una millora de la comoditat a l'hora d'utilitzar els espais on es duen a terme els diferents esdeveniments; evitant possibles solapaments i errors en la gestió de les reserves.

Aquest treball està centrat principalment en el programa, en el tractament de les dades extretes del calendari i en la gestió dels horaris de les aules per proporcionar un servei clar i ordenat.

A més a més, es dona un seguit de pautes de com utilitzar el sistema, de les possibilitats que ofereix i del significat dels elements (paraules o símbols) que apareixen en pantalla.

Sumari

RESUM	3
SUMARI	4
1. PREFACI	6
1.1. Origen del projecte.....	6
1.2. Motivació	6
2. INTRODUCCIÓ	7
2.1. Objectius del projecte.....	7
2.2. Abast del projecte	7
3. REQUERIMENTS I SELECCIÓ DE MATERIALS	8
3.1. Selecció de la placa del sistema	8
3.2. Selecció de la pantalla del sistema	9
3.2.1. Explicacions tècniques	10
4. IMPLEMENTACIÓ DELS COMPONENTS	13
5. PROGRAMACIÓ	16
6. INSTRUCCIONS D'ÚS	17
7. PRESSUPOST	20
CONCLUSIONS	21
POSSIBLES MILLORES	22
AGRAÏMENTS	23
BIBLIOGRAFIA	24
Referències bibliogràfiques	24
ANNEXOS	25
Annex 1: Codi de programació Arduino	25
Annex 2: Script de Google: pantalla secundària	32
Annex 3: Script de Google: pantalla principal.....	34

1. Prefaci

1.1. Origen del projecte

Tenir un espai compartit dins d'un entorn de treball, al qual poden accedir diverses persones i per motius diferents, fa que càpiga la possibilitat que, en certes ocasions, no s'efectuïn les reserves de forma adequada o simplement sorgeixi una situació imprevista que requereixi de l'ús d'aquest espai. Aquestes són causes que poden generar petits conflictes que desemboquin en malentesos degut a que ens trobem davant d'una situació on dues persones sol·liciten fer ús d'una sala en el mateix interval de temps. El conflicte apareix quan, degut a aquests motius, es generen solapaments o una persona es troba ocupada l'aula que havia reservat ja que, la persona que l'està utilitzant l'havia trobat buida i havia decidit entrar-hi per fer-ne ús.

És en aquest moment on pren sentit la idea de trobar una solució per poder evitar aquests problemes. Tant és així, que després de pensar en la manera més eficient de resoldre aquests conflictes i aprofitant els avenços tecnològics que tenim a l'abast, es va decidir desenvolupar aquest treball.

Com a solució es va optar per la instal·lació d'unes pantalles a l'entrada de les sales en qüestió que mostressin les properes reserves i evitessin malentesos innecessaris gràcies a l'aclariment de qualsevol mena de dubte sobre l'activitat que es duria a terme en cada moment en aquell espai.

1.2. Motivació

El motiu principal pel qual he escollit aquest tema com a treball de fi de grau ha sigut el meu interès en l'electrònica però també la meva inquietud a l'hora d'aprendre nous conceptes i llenguatges de programació. El fet de poder implementar-los i treure'n profit en possibles projectes al llarg de la meva vida professional ha sigut també un factor rellevant per la selecció de la temàtica.

Quan se'm va presentar la possibilitat de treballar aquest projecte vaig considerar que era una oportunitat interessant per poder adquirir nous coneixements i alhora un repte.

2. Introducció

2.1. Objectius del projecte

Aquest treball tracta de l'organització i gestió d'un calendari de reserves d'un seguit d'aules i sales per facilitar i millorar l'experiència dels usuaris que en fan ús.

L'objectiu d'aquest projecte és la implementació i programació d'un microcontrolador amb la finalitat que, a través de les corresponents peticions als servidors de Google, adquireixi la informació donada per un calendari i gestionar-la per poder plasmar-la en una pantalla ubicada a l'entrada de cadascuna de les aules.

Aquest treball pretén ajudar en la gestió d'horaris d'un departament de l'Escola; concretament, evitant l'aparició de conflictes a l'hora de fer ús de les aules a conseqüència del desconeixement de l'ocupació d'aquests espais i de les reserves fetes prèviament.

2.2. Abast del projecte

El treball consisteix en el desenvolupament d'un prototipus format, principalment, per una pantalla i un microcontrolador. Aquests elements es preveu que se situïn a l'accés d'un seguit de sales, on poder visualitzar si estan ocupades o lliures i obtenir informació referent a propers esdeveniments que hi tindran lloc.

Per simplificar el projecte, només es realitzarà la programació d'una aula genèrica. Aquesta decisió és deguda que es considera que, quan finalment s'implementa aquest sistema en més aules, la reproducció es podrà aconseguir de forma senzilla. Fent petites modificacions en el codi es podrà aconseguir personalitzar cada pantalla i que faci referència a l'aula en la qual està instal·lada. Malgrat això, és important poder mostrar en cada pantalla la informació de la resta d'aules per facilitar l'ús de les instal·lacions.

3. Requeriments i selecció de materials

Primerament, es presenten els requeriments principals del projecte i s'ha dut a terme una breu investigació per escollir els components que millor encaixen amb la naturalesa d'aquest.

Les característiques que ha de complir el sistema són les següents:

- Tenir un xip que permeti connexió Wi-Fi
- La placa escollida ha d'assegurar els protocols de seguretat per accedir als servidors de Google
- La placa ha de permetre connectivitat amb una pantalla
- El xip de processament ha de tenir una capacitat major a la demanda màxima requerida pel sistema
- La pantalla ha de ser de mida petita

La primera fase del treball ha consistit a fer una cerca dels materials que compleixen aquestes característiques per poder desenvolupar-lo de forma correcta [2] [4] [5].

3.1. Selecció de la placa del sistema

L'element principal del sistema és la placa i el corresponent xip que es programarà posteriorment. És per aquest motiu que ha sigut el començament de l'estudi. S'han seleccionat les que s'han considerat les dues plaques més adients pel projecte de marques diferents que semblen una bona opció dins del mercat.

Raspberry Pi 3

Les plaques de la marca Raspberry són conegudes per tenir un bon rendiment a causa de la seva potència de processament.

La Raspberry Pi 3 és un model que compleix les característiques requerides pel nostre projecte, ja que té una mida petita, una capacitat de processament molt superior a la necessària i permet connexió via Wi-Fi.

Un altre aspecte molt important a tenir en compte és el procés d'enciptació de la informació per poder comunicar-se amb els servidors de Google i ser capaç d'obtenir la informació del calendari per tractar-la posteriorment i mostrar-la al display.

El punt negatiu que trobem en aquesta placa i ens fa descartar-la és l'elevat preu de mercat que

té a causa de les seves prestacions.

Arduino MKR Wi-Fi 1010

Aquesta placa destaca per la seva mida, la seva capacitat de processament i a les prestacions Wi-Fi que ofereix.

També està constituïda per un xip d'encriptació (anomenat ATECC508A) que ofereix la seguretat suficient perquè les peticions realitzades als servidors de Google siguin acceptades sense cap mena de problema.

Destacar també que el controlador que permet la connexió sense fils s'anomena NINA i el microprocessador que porta incorporat és un ESP32, oferint una capacitat de processament de 32 bits. Els dos elements han sigut dissenyats amb unes prestacions que s'adapten correctament al projecte.

Degut a les comparacions fetes i a les característiques comentades, s'ha decidit optar per la placa d'Arduino. Mencionar que les dues plaques utilitzen llenguatges de programació diferents, alguns dels quals s'han estudiat a l'escola per entendre el funcionament i poder treballar el desenvolupament de codi.

3.2. Selecció de la pantalla del sistema

Un cop triada la placa que constitueix el nucli del sistema, és adequat decidir quina pantalla és compatible i s'adapta millor a aquesta.

El següent pas que s'ha realitzat ha sigut accedir a un dissenyador, fabricant i proveïdor anomenat Adafruit per observar i comparar l'ampli mercat existent en l'entorn d'Arduino. D'entre tots els productes se n'han seleccionat tres compatibles amb la placa escollida i que semblen adients per a la realització del projecte.

Abans d'entrar en detall amb els diferents models de pantalla que s'han seleccionat, s'han volgut fer un seguit d'explicacions tècniques per facilitar la comprensió de les consideracions i decisions fetes en aquest punt del treball.

3.2.1. Explicacions tècniques

Abans de continuar, és interessant comentar un seguit de característiques o diferències tècniques que han ajudat a realitzar les decisions que es comenten més endavant [2] [3].

Diferències entre una pantalla TFT i una LCD

Una pantalla LCD (*liquid crystal display*) és un monitor que utilitza les propietats de modulació lumínica dels cristalls líquids combinats amb filtres polaritzats. Aquests dispositius no emeten llum pròpia sinó que s'ajuden d'una tècnica anomenada retroil·luminació, la qual es basa en una font de llum integrada que permet millorar el contrast de visualització i assegurar una bona llegibilitat.

Dins d'aquest grup de pantalles hi trobem les TFT (*thin film transistor*), que usen un tipus especial de transistors d'efecte camp que es fabriquen dipositant fines pel·lícules d'un semiconductor actiu (per exemple silici), així com capes de material dielèctric i contactes metàl·lics sobre un suport (per exemple vidre).

Diferències entre un sensor tàctil capacitiu i un de resistiu

Una pantalla tàctil resistiva està formada per dues capes primes de material separades per un espai. La capa superior sol ser de policarbonat, mentre que la inferior està feta d'un material rígid. Es basen en la tecnologia de variació de pressió; quan es prem la pantalla, es crea un canvi de resistència mecànica i això fa que el sensor detecti aquesta variació i un processador calculi les coordenades on s'ha produït la pulsació.

Per altra banda, les pantalles tàctils capacitives, es troben constituïdes per un material transparent i conductor cobert per una làmina de vidre. També consten d'elèctrodes que mantenen un voltatge al llarg de tota la pantalla. Quan toquem aquest vidre, el sistema s'aprofita que el cos humà és un conductor natural i inicia un flux d'energia entre els elèctrodes i el nostre dit permetent que els sensors detectin un canvi de voltatge i ubiquin on s'ha produït la pulsació.

Un cop aclarides les diferències existents entre aquests conceptes, és adient analitzar en detall les diferents opcions seleccionades per escollir quina serà la pantalla del sistema d'aquest projecte.

Pantalla LCD estàndard Adafruit 20 x 4, model 198

La primera pantalla seleccionada s'ha introduït dins d'aquesta classificació principalment per dos motius. La seva petita mida i sobretot la seva senzillesa a l'hora d'interactuar amb ella són

característiques destacables. Per altra banda cal tenir en compte que l'objectiu és poder mostrar informació d'una agenda de forma entenedora i visual. Aquesta pantalla té una mida de visualització força petita i això podria ser un inconvenient quan es vulgui mostrar el contingut. Mencionar també el preu assequible d'aquest element.

Pantalla TFT Adafruit de 2,8 polsades amb sensor tàctil capacitiu, model 1947

Aquesta opció destaca per ser una pantalla amb una qualitat alta comparant-la amb d'altres de gamma similar. El sensor tàctil aporta un afegit que pot ser molt positiu a l'hora d'implementar i utilitzar el sistema i una altra de les especificacions que compleix és que té una mida petita.

Degut a les característiques mencionades, aquesta pot ser una molt bona opció a escollir per desenvolupar el projecte. Malgrat això, s'ha de tenir en compte que al ser un dispositiu de major qualitat, el seu preu augmenta, sent aquest un punt força negatiu.

Pantalla TFT Adafruit de 2,8 polsades amb sensor tàctil resistiu, model 1651

L'última opció escollida ha sigut un model molt similar a la segona opció però amb unes característiques inferiors. També consta d'un sensor tàctil i la pantalla és de la mateixa mida que l'anterior opció. La qualitat d'imatge és una mica pitjor, ja que es poden apreciar els píxels dels quals està formada però s'ha de tenir present que la informació que es mostrarà en pantalla són principalment caràcters de l'abecedari i figures geomètriques.

Un cop seleccionades les diferents alternatives compatibles amb el microcontrolador escollit, s'ha fet una valoració de les característiques que s'han considerat més importants referents a la pantalla.

El fet de comptar amb un sensor tàctil aporta la possibilitat de permetre navegar entre diferents pantalles, podent mostrar així més contingut que pugui ajudar als usuaris del sistema a obtenir més informació i a interaccionar amb el propi dispositiu.

Un altre aspecte a valorar és, com s'ha mencionat prèviament, la mida. La pantalla ha de ser prou petita per poder situar-la al costat d'un accés sense suposar un entrebanc i prou gran perquè qualsevol persona pugui llegir la informació que s'hi mostra de forma clara i entenedora amb un sol cop d'ull.

Per aquests motius s'ha decidit que la pantalla escollida per formar part del projecte és la tercera opció.

Deixant de banda aquests dos elements, s'ha volgut donar més presència i forma al projecte incorporant un dispositiu detector de moviment que ajudi a estalviar energia i a interaccionar amb l'usuari. En aquest cas no s'ha fet una investigació tan exhaustiva com amb els altres dispositius ja que és un element que té un cost menor i la seva funcionalitat és senzilla, permetent que els diferents models siguin molt similars entre si.

4. Implementació dels components

A continuació es mostra un esquema de les connexions entre els diferents dispositius del sistema:

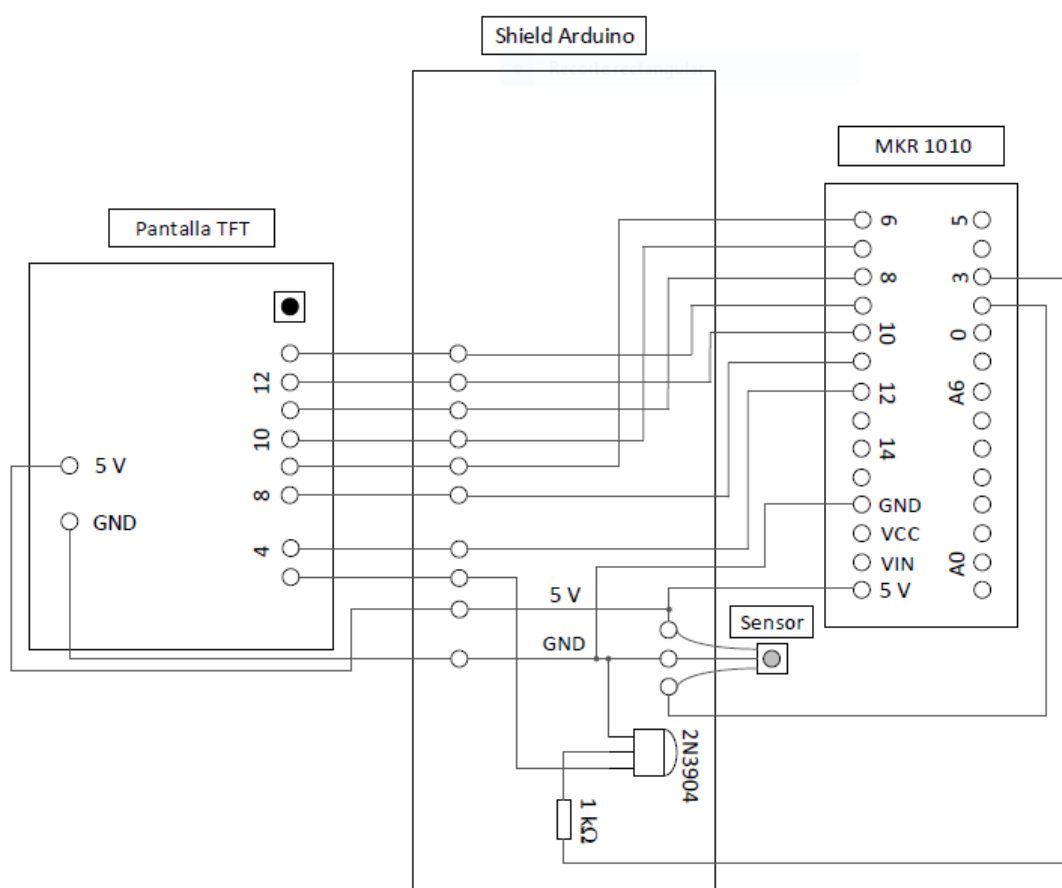


Figura 1.1 - Esquema de connexions (font pròpia)

Per facilitar la lectura d'aquest esquema i fer-lo més entenedor, s'ha realitzat una simplificació amb la finalitat de destacar la informació més rellevant (Figura 1.2).

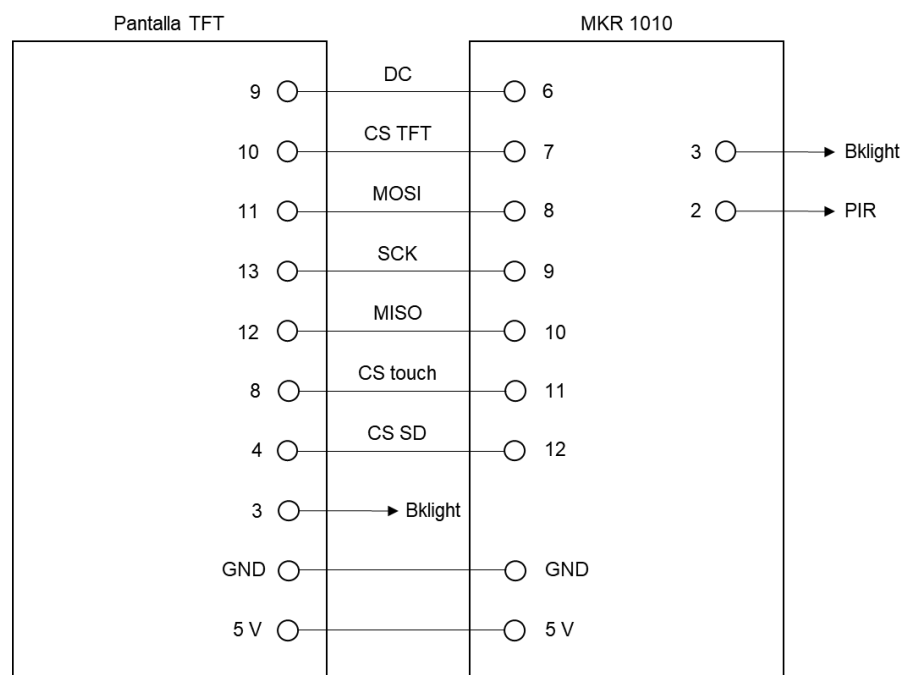


Figura 1.2 - Esquema de connexions simplificat (font pròpia)

La funció del *shield* és facilitar aquestes connexions i permetre la implementació d'altres components.

La connexió entre la pantalla i la placa s'anomena SPI (*serial peripheral interface*) i es tracta d'un protocol utilitzat pels microcontroladors per poder comunicar-se amb perifèrics o altres microcontroladors de forma ràpida i a curta distància. En aquest tipus de connexió sempre hi ha un element mestre que controla altres dispositius (anomenats esclaus) i tres connexions comunes en tots els dispositius que utilitzen protocol SPI:

- MISO (*master in slave out*) – Línia a través de la qual l'esclau pot enviar informació al dispositiu mestre.
- MOSI (*master out slave in*) – Línia a través de la qual el mestre pot enviar dades als perifèrics.
- SCK (*serial clock*) – Relloige de polsos generats pel mestre que sincronitza la transmissió d'informació.

També cal destacar que la pantalla té una pota de retroil·luminació que encén la llum perquè sigui visible el contingut mostrat en aquesta. És important tenir present que aquesta pota s'activa

quan es posa a nivell baix.

Per altra banda, convé mencionar que el sensor de moviment utilitza una connexió molt senzilla, amb tres cables per alimentació i dades.

La resistència i el transistor s'han afegit de forma addicional per seguretat i per compensar la diferència de voltatge entre els elements. La placa funciona amb un voltatge de 3,3 V i és així com s'ha d'alimentar, però permet la possibilitat de donar una sortida a 5 V a altres elements que ho necessitin, com és el cas de la pantalla tàctil.

A la Taula 1 es mostra un resum de la funció assignada a les potes de la placa MKR.

Pota	2	3	6	7	8	9	10	11	12
Connexió	PIR	BKlight	DC	CS TFT	MOSI	SCK	MISO	CS touch	CS SD

Taula 1 - Definició de potes de la placa (font pròpia)

Entrant en detall en aquest apartat de connexions, cal mencionar que les potes 2, 7, 11 i 12 són CS (*chip select*); són potes que s'utilitzen com a referència perquè el microcontrolador sàpiga en cada moment amb qui intercanvia informació. També és important saber que la pota número 3 s'utilitza per encendre i apagar la retroil·luminació de la pantalla. Per últim, cal mencionar que la pota 6 s'utilitza per transmetre i comunicar informació i s'anomena DC (*data/command select*).

5. Programació

Aquesta part ha sigut la més extensa del projecte i a la que se li ha dedicat més temps. En aquest cas, el codi de programació s'ha basat en un llenguatge propi d'Arduino que realment és una adaptació simplificada de C++.

Per poder arribar a programar el microcontrolador perquè realitzi les funcions desitjades, s'ha fet una recerca prèvia per entendre conceptes més avançats sobre el llenguatge utilitzat. Un cop obtinguda aquesta informació, s'ha creat el que ha acabat sent el codi final a través de diferents segments, assegurant el seu correcte funcionament. Cal mencionar que algunes parts de la programació s'han basat en altres codis (trobat per internet o cedits pel tutor del treball [1] [6], ja sigui com a idea o a nivell estructural.

Per altra banda, també s'ha treballat en uns *scripts* de Google que donen com a resultat un seguit de dades del calendari, com per exemple, el nom de l'esdeveniment i l'hora a la qual està previst que tingui lloc.

Per aconseguir integrar els dos sistemes i que les peticions fetes pel microcontrolador siguin acceptades pels servidors de Google i aquests enviïn la informació desitjada, s'ha dedicat una part del codi a assegurar aquesta comunicació tenint present que, a l'hora de fer una petició als servidors, aquests realitzen un redireccionament per temes de seguretat.

Un cop obtinguda la informació dels scripts, aquesta s'analitza i es processa de forma adient per, posteriorment, ser mostrada de forma clara i precisa a la pantalla.

Per últim, s'ha tingut en compte el sensor de presència com a mètode per estalviar energia. La idea principal és que quan una persona estigui mirant la pantalla o hi passi per davant, aquesta s'il·lumini, es mantingui encesa un temps determinat i es torni a apagar, entrant en una espècie de mode repòs però sense que el microcontrolador deixi de treballar, deixant la pantalla sense retroil·luminació, ja que és l'element que més energia consumeix de tot el sistema.

És important comentar que s'han realitzat uns annexos on s'expliquen en detall les diferents parts del codi de programació. Si es vol observar detingudament com s'ha creat i quina funció fa cada bloc del codi, els annexos es troben al final d'aquest document.

6. Instruccions d'ús

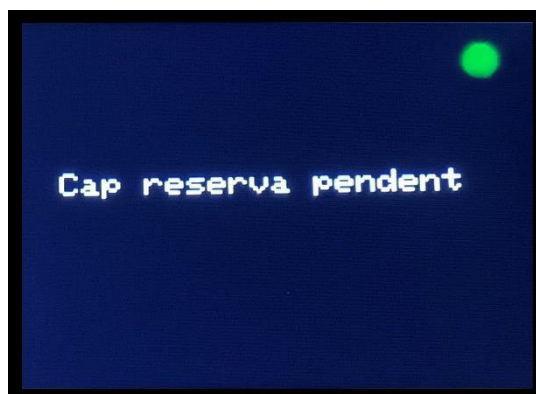
Com s'ha mencionat prèviament en diverses ocasions, el sistema utilitzat en aquest projecte consta de dues parts principals: la pantalla i el microcontrolador. Aquest últim s'ha programat basant-se en un seguit d'idees i complint en tot moment una de les premisses principals que consisteix a mostrar les dades que s'han considerat més significatives de forma clara i precisa en la visualització en pantalla.

En aquest apartat s'han inclòs un seguit d'instruccions per facilitar i entendre el funcionament del sistema. L'usuari només es podrà comunicar amb aquest a través del sensor tàctil de la pantalla.

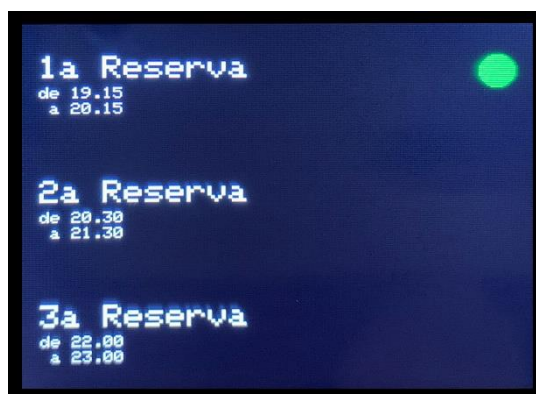
Degut a les característiques del projecte, s'ha decidit separar la informació en dos blocs; un personalitzat amb les dades corresponents a la sala a la qual la pantalla ha de fer referència, és a dir, on es troba instal·lada. Mentre que el segon bloc mostra la informació referent al conjunt de totes les sales que el Departament utilitza per portar a terme els esdeveniments.

Aquesta decisió de segmentar les dades en dos grans grups, s'ha pres amb l'objectiu de facilitar la lectura en pantalla i que aquesta sigui més clara.

Normalment, ens trobarem el sistema en una primera pantalla, que anomenarem pantalla principal (Il·lustracions 1 i 2). Aquí podrem observar els tres propers esdeveniments que tindran lloc a l'aula en la qual ens trobem, si és que n'hi ha tants de previstos. Com a informació addicional també tindrem un cercle que canvia de color en funció de l'estat de la sala. Si aquesta està ocupada, el cercle serà de color vermell, mentre que si està lliure, serà de color verd. Si l'espai al qual volem accedir està lliure (cercle de color verd), cal observar quan comença l'esdeveniment pròxim per tal d'evitar conflictes amb altres usuaris que hagin reservat l'aula. A més a més, en aquesta pantalla principal, podrem prémer la cantonada superior dreta (ubicació del cercle comentat) per tenir accés a una segona pantalla, la qual anomenarem pantalla secundària.

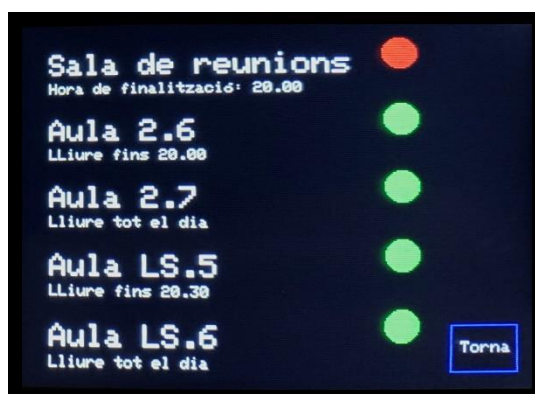


Il·lustració 1 - Pantalla principal sense reserves (font pròpia)



Il·lustració 2 - Pantalla principal (font pròpia)

Un cop ens trobem en aquesta segona pantalla (Il·lustració 2), podrem observar la informació referent a l'estat de cadascuna de les cinc aules de les quals el departament disposa per fer reserves (inclosa la corresponent a la pantalla). La distribució d'informació en la pantalla secundària consisteix en el nom de les aules i, sota d'aquest, el seu estat (lliure tot el dia, lliure fins a una hora determinada o ocupada i amb l'hora de finalització de l'esdeveniment actual). Seguint la metodologia de la pantalla principal, al costat del nom de cada sala hi haurà un cercle de color verd o vermell seguint el criteri explicat anteriorment.



Il·lustració 3 - Pantalla secundària (font pròpia)

Troband-nos a la pantalla secundària, si l'usuari vol tornar a la pantalla anterior haurà de prémer un botó rectangular situat a la cantonada inferior dreta on hi ha escrita la paraula "Torna". Si no ho fa, un cop passat un període de temps determinat, el microcontrolador retorna automàticament a la pantalla per defecte corresponent a la sala que pertoca.

Un dels problemes que es vol evitar són els solapaments, de tal manera que quan hi ha dos esdeveniments que comencen a la mateixa hora, l'últim pel qual s'ha fet la reserva, apareixerà situat sobre el que realment tingui prioritat respecte a l'aula. Fent referència a la il·lustració 2, quan es produeixi solapament, l'esdeveniment per al qual s'havia reservat l'aula en primera

instància, apareixerà al centre de la pantalla, mentre que l'esdeveniment que genera el conflicte es visualitzarà a la part superior d'aquesta. D'aquesta manera, podem observar que el que aparegui en una zona inferior, és l'esdeveniment que hi tindrà lloc en cas de discussió.

Un aspecte a tenir en compte és que el microcontrolador té una capacitat limitada que fa que la velocitat de resposta també ho sigui. Això es tradueix en la rapidesa que té el sistema per fer les peticions corresponents a Google, executar el codi de programació i mostrar la informació desitjada en pantalla. Per evitar-li problemes al microcontrolador a l'hora de processar la informació s'ha fet que quan aquest ja ha fet una petició als servidors i està treballant amb les dades extretes d'aquests, apareguin punts suspensius a la part inferior dreta de la pantalla per tal que l'usuari entengui que s'ha de tenir paciència fins que desapareguin i poder veure la informació actualitzada o interactuar amb el sistema.

Cal mencionar també que quan es canvia de la pantalla principal a la secundària o viceversa, el canvi no és immediat i apareix un text en pantalla que indica que la informació s'està processant; aquestes pantalles de càrrega on apareix aquest text, que funcionen com a intermediàries entre els dos blocs d'informació mencionats, es visualitzaran en pantalla durant un període curt de temps.

7. Pressupost

Per últim, s'indica el preu dels elements que s'han utilitzat per realitzar aquest projecte. Cal mencionar que és orientatiu i només s'ha considerat el desenvolupament d'una sola unitat del sistema. No s'hi ha inclòs les hores i la feina de programació del codi; merament s'ha tingut en compte el material utilitzat (excloent despeses d'enviament).

Material	Preu (€)
Placa Arduino MKR WiFi 1010	27,90
Pantalla TFT Adafruit	31,26
Sensor PIR Adafruit	8,90
Resistència, transistor i cablejat	0,30
PREU TOTAL	68,36

Taula 2 - Pressupost (font pròpia)

Conclusions

Un cop realitzat el treball s'ha pogut demostrar que, amb el material escollit, ha sigut possible crear un sistema que gestioni les dades extretes d'un calendari a través de peticions fetes als servidors de Google. Gràcies a això i a la programació del microcontrolador, s'ha aconseguit la visualització de l'esmentada informació en les pantalles situades a l'accés de les sales determinades pel Departament.

Primerament, cal destacar que s'ha complert l'objectiu principal del projecte, que era muntar i programar un sistema que prestés la funció d'ajudar a evitar mal entesos i millorar l'experiència a l'hora d'utilitzar els espais on se situen les pantalles.

Per altra banda, és important comentar que el treball de fi de grau ha sigut una motivació per estudiar gran part del funcionament del llenguatge de programació que utilitzen els microcontroladors implementats a les plaques fabricades per Arduino.

Possibles millores

El món tecnològic es troba en una constant evolució. Això fa que si un sistema electrònic es vol mantenir actualitzat, s'hagi d'anar millorant i fent evolucionar a mesura que canvien les necessitats dels usuaris que en fan ús.

Si en un futur, es volgués millorar el sistema del projecte, una manera d'aconseguir-ho seria millorar la qualitat de les pantalles instal·lades tot i que les que s'han utilitzat per a aquest projecte compleixen la seva funció de forma correcta; seria una millora en la qualitat visual que són capaces d'aportar. Fent referència a aquestes pantalles, una bona opció seria fer un estudi per optimitzar la informació que es mostra en aquestes, ja que és requerida o important per als usuaris.

Per altra banda, fent referència al microcontrolador, es podria fer una inversió per substituir-lo per un altre dotat per una capacitat de processament superior a l'actual. Malgrat això, aquest element es troba limitat per la connexió a internet i la velocitat de resposta dels servidors.

Una manera d'estalviar encara més energia seria definir dues franges horàries en les quals el bucle del codi d'Arduino fes les peticions i refresqués les pantalles amb una freqüència diferent en funció de si l'Escola es troba oberta o tancada.

Un cop mencionats aquests detalls, caldria comentar que el projecte compleix la funció per la qual ha estat dissenyat i, per tant, les millores que s'hi poden trobar són a nivell qualitatiu dels propis elements o degut als criteris dels usuaris que en fan ús de forma habitual.

Agraïments

Pel fet que aquest projecte significa el final d'una etapa important en la meua vida universitària i personal, m'agradaria fer un seguit de mencions per reconèixer la gent que sap l'esforç, treball i ganes que hi ha darrere d'aquests anys.

En primer lloc, voldria donar les gràcies al que ha sigut el tutor d'aquest projecte, Oriol Boix. La vocació que mostra per la seva professió ha estat present en cada cita o correu per parlar sobre qualsevol tema relacionat amb el treball, amb l'entorn tecnològic i de l'Escola. L'atenció i l'interès mostrats en tot moment han fet que fos molt fàcil treballar amb ell i m'han ajudat a aclarir qualsevol dubte que hagi sorgit durant el desenvolupament del projecte.

Per altra banda, m'agradaria agrair a la meua família, parella i amics per donar-me suport en tot moment i ajudar-me a superar els obstacles que han pogut sorgir durant el transcurs del grau.

Bibliografia

Referències bibliogràfiques

- [1] RECURSOS CITCEA [<https://recursos.citcea.upc.edu>, 2019] Data de consulta: 16 de Setembre, 2019.
- [2] ARDUINO [<https://www.arduino.cc>, 2019] Data de consulta: 16 de Setembre, 2019.
- [3] ARDUINO. CURSO PRÁCTICO DE FORMACIÓN, Óscar Torrente Artero, RC Libros, 2013. Data de consulta: 17 de Setembre, 2019.
- [4] PROGRAMAR FÁCIL [<https://programarfacil.com>, 2019] Data de consulta: 19 de Setembre, 2019.
- [5] GITHUB [<https://github.com/mobizt/Firebase-Arduino-WiFiNINA>, 2019] Data de consulta: 21 de Setembre, 2019.
- [6] INSTRUCTABLES [<https://www.instructables.com/id/Irrigation-Using-Google-Calendar>, 2017] Data de consulta: 23 de Setembre, 2019.
- [7] ESP8266 COMMUNITY WIKI [<https://www.esp8266.com/wiki/doku.php>, 2019] Data de consulta: 25 de Setembre, 2019.
- [8] S-TOUCH®: ADVANCED TOUCHSCREEN CONTROLLER [<https://cdn-shop.adafruit.com/datasheets/STMPE610.pdf>, 2011] Data de consulta: 4 de Novembre, 2019.
- [9] A-SI TFT LCD SINGLE CHIP DRIVER [<https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf>, 2010] Data de consulta: 4 de Novembre, 2019.

Annexos

Annex 1: Codi de programació Arduino

```
// Aquest programa està parcialment basat en els exemples de les pàgines
// https://www.arduino.cc/en/Tutorial/LibraryExamples#wifi1010
// https://recursos.citcea.upc.edu/iot/exemple/

#include <SPI.h>      // Carreguem la llibreria SPI
#include <WiFiNINA.h>  // Carreguem la llibreria WiFiNINA
#define server_len 50
#define pag_len 400
#define mis_len 5     // Nombre màxim de línies de la resposta
#include "arduino_secrets.h" // Info en una altra pestanya per seguretat
//Inicialització pantalla tft
#include "Adafruit_GFX.h"    // Llibreria gràfica
#include "SPI.h"
#include "Wire.h"
#include "Adafruit_ILI9341.h"
#include "Adafruit_STMPE610.h"
// Definim la informació per calibrar el sensor tàctil
#define TS_MINX 150
#define TS_MINY 130
#define TS_MAXX 3800
#define TS_MAXY 4000
// El STMPE610 utilitza una connexió hardware SPI
#define STMPE_CS 11
Adafruit_STMPE610 ts = Adafruit_STMPE610(STMPE_CS);
// El display també utilitza una connexió SPI
#define BkLig 3      // Llum pantalla
#define TFT_DC 6
#define TFT_CS 7
#define SD_CS 12
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
//Connexió a la xarxa
const char idXarxa[] = SECRET_SSID;      // Nom del punt d'accés
const char contrasenya[] = SECRET_PASS;  // Contrasenya de connexió
char server[server_len];
char pagina[pag_len];
unsigned long darreraConnexio = 0;
const unsigned long periodeConnexio = 10000UL;
bool pendent, completa, redir;
bool ara = false;
bool ini_msg = false;
int comp_lin = 0;
String peticio = "";      // Aquí guardarem una línia de la petició del client
String peticioAux = "";   // la petició anterior (també ho farem servir de reserva)
String missatge[mis_len]; // Aquí hi guardarem el text rebut
String server1, paginal;
int status = WL_IDLE_STATUS;
WiFiSSLClient client;
//pantalla 240x320pixels
int altura_linia = 20;
#define sensorpir 2
String missatgeAnterior[mis_len];
bool ocupat;
bool pantallaSec = false;
bool pantallaSecOK = false;
bool capTasca; // Refresca pantalla quan "Cap reserva pendent"
bool pensant;
int contTorna = 0; // Esperar X "loops" per tornar de la pantalla secundària a la principal
```

```

int contPir = 0;
#define periodeEnces 60000UL    // 60 s
unsigned long darrerEnces = 0;

void processa(String missat) {
    String Acte, Inici, Final, Estat;    // Aquí guardarem les dades
    Acte = missat.substring(0, missat.indexOf(","));
    missat = missat.substring(missat.indexOf(",") + 1);
    Inici = missat.substring(0, missat.indexOf(","));
    missat = missat.substring(missat.indexOf(",") + 1);
    Final = missat.substring(0, missat.indexOf(","));
    Estat = missat.substring(missat.indexOf(",") + 1);
    if (!pantallaSec) {
        Serial.print("Acte: ");
        Serial.println(arreglaEsp(Acte));
        Serial.print("Inici: ");
        Serial.println(Inici);
        Serial.print("Final: ");
        Serial.println(Final);
        tft.setTextColor(ILI9341_WHITE);
        tft.setCursor(20, altura_linia);
        if (Acte.length() > 18) {
            Acte = Acte.substring(0, 18);
        }
        tft.println(arreglaEsp(Acte));
        tft.setCursor(20, altura_linia + 20);
        tft.setTextSize(1);
        tft.print(F("de "));
        tft.println(Inici);
        tft.setCursor(70, altura_linia + 20);
        tft.print(F(" a "));
        tft.println(Final);
        tft.setCursor(120, altura_linia + 20);
        tft.print(F("h"));
        tft.setTextSize(2);
        if (Estat == "0") { // 0 vol dir "En curs"
            ocupat = true;
        }
    }
    if (pantallaSec) {
        tft.setTextColor(ILI9341_WHITE);
        tft.setCursor(20, altura_linia);
        tft.println(arreglaEsp(Acte));
        if (Estat == "0") { // 0 vol dir "En curs"
            ocupat = true;
            tft.fillCircle(240, altura_linia, 10, ILI9341_RED);
            tft.setTextSize(1);
            tft.setCursor(20, altura_linia + 20);
            tft.print(arreglaEsp("Hora de finalització: "));
            tft.println(Final);
            tft.setCursor(185, altura_linia + 20);
            tft.print(F("h"));
            altura_linia += 5;
            tft.setTextSize(2);
        }
        if (Estat == "1") { // 1 vol dir "Lliure"
            ocupat = false;
            tft.fillCircle(240, altura_linia, 10, ILI9341_GREEN);
            tft.setTextSize(1);
            tft.setCursor(20, altura_linia + 20);
            tft.print(F("Lliure fins "));
            tft.println(Inici);
            tft.setCursor(125, altura_linia + 20);
            tft.print(F("h"));
        }
    }
}

```

```

        altura_linia += 5;
        tft.setTextSize(2);
    }
    if (Estat == "2") { // 2 vol dir "Lliure tot el dia"
        ocupat = false;
        tft.fillCircle(240, altura_linia , 10, ILI9341_GREEN);
        tft.setTextSize(1);
        tft.setCursor(20, altura_linia + 20);
        tft.print(F("Lliure tot el dia"));
        altura_linia += 5;
        tft.setTextSize(2);
    }
}
}

// Variables per definir correctament el text a pantalla
static const char especials[] PROGMEM = { -73, -128, -121, -120, -119, -115, -113, -110,
-109, -102, -100, -96, -89, -88, -87, -83, -81, -78, -77, -70, -68};
static const char substituïts[] PROGMEM = { -7, 31, -128, 30, -112, 29, -117, 28, 27, 26,
-102, -123, -121, -118, -126, -95, 25, -107, -94, -93, -127};

// Funcions per definir correctament el text a pantalla
String arreglaEsp(String text) {
    String nouTxt = "";
    for (byte j = 0; j < text.length(); j++) {
        char carac = text[j];
        if ((carac != -61) && (carac != -62) && (carac != 194) && (carac != 195)) {
            if ((carac < 0) or (carac > 127)) {
                char nou = '?'; // Si no és cap dels previstos tornarà un interrogant
                for (byte i = 0; i < sizeof(especials); i++) {
                    if (carac == especials[i]) {
                        nou = substituïts[i];
                    }
                }
                carac = nou;
            }
            nouTxt = nouTxt + carac;
        }
    }
    return nouTxt;
}

void setup() { // Inicialització
    pinMode(sensorpir, INPUT_PULLUP);
    pinMode(BkLig, OUTPUT);
    Serial.begin(9600); // Monitor sèrie
    tft.begin();
    tft.setRotation(1);
    ts.begin();
    while (!Serial) {
        ; // Esperem que l'usuari obri el monitor sèrie
    }
    if (WiFi.status() == WL_NO_MODULE) {
        Serial.println(F("No s'ha trobat el dispositiu Wi-Fi"));
        // La funció F obliga al compilador a guardar el text a la memòria de programa
        while (true); // Bloquegem el programa
    }

    String versio = WiFi.firmwareVersion(); // Necessitem una actualització de software?
    if (versio < "1.0.0") {
        Serial.println(F("Convindria actualitzar el firmware"));
        tft.print(F("Convindria actualitzar el firmware"));
    }
}

```

```

while (status != WL_CONNECTED) { // Connexió a la xarxa wifi
  Serial.print(F("Connectant a la xarxa "));
  Serial.println(idXarxa);
  status = WiFi.begin(idXarxa, contrasenya);
  tft.setTextSize(2);
  tft.fillScreen(ILI9341_BLACK);
  tft.setTextColor(ILI9341_WHITE);
  tft.print(F("Connectant a la xarxa "));
  tft.setCursor(20, 20);
  tft.println(F(idXarxa));
  delay(10000); // Ho tornarem a intentar passats 10 s
}
Serial.print(F("Connectat a "));
Serial.println(WiFi.SSID());
Serial.print(F("Estat de la connexió: "));
Serial.println(WiFi.status());
Serial.print(F("Adreça IP del dispositiu: "));
Serial.println(WiFi.localIP());
Serial.print(F("Intensitat del senyal: "));
Serial.print(WiFi.RSSI());
Serial.println(F(" dBm"));
Serial.println();
Serial.println(F("Anem a connectar amb el servidor"));
redir = false;
tft.fillScreen(ILI9341_BLACK);
tft.setCursor(20, 20);
tft.setTextColor(ILI9341_WHITE);
tft.print(F("Connectant amb el"));
tft.setCursor(20, 40);
tft.print(F("servidor..."));

}

void loop() { // Programa que es repeteix indefinidament
  // El bucle principal té tres parts:
  // 1. Gestió dels caràcters que arriben
  // 2. Tractament de les dades rebudes
  // 3. Nova petició quan ha passat el temps
  altura_linia = 20;

  if (pensant) { // Evita interacció amb pantalla quan s'està fent petició
    tft.setCursor(300, 230);
    tft.setTextSize(1);
    tft.setTextColor(ILI9341_WHITE);
    tft.print(F("..."));
    tft.setTextSize(2);
  }

  if (!pensant) { // Torna a concedir interacció de l'usuari amb la pantalla
    tft.setCursor(300, 230);
    tft.setTextSize(1);
    tft.setTextColor(ILI9341_BLACK);
    tft.print(F("..."));
    tft.setTextSize(2);
  }

  while (client.available()) {
    // Gestió dels caràcters que arriben
    // Aquest bucle va guardant els caràcters rebuts
    // i espera al moment en que arriba un salt de línia
    char c = client.read(); // Rebem caràcters del servidor
    if (c == '\n') { // Mirem si és un salt de línia
      peticioAux = peticio; // Guardem la petició anterior
      peticio = ""; // Ens preparem per a la línia següent
    }
  }
}

```

```

    completa = true;    // Preparat per tractar-ho
} else {
    peticio += c;    // Afegim el caràcter rebut
}
// Quan ha arribat un salt de línia, hem de mirar què ha arribat
if (completa) { // Ha arribat una línia completa
    // El nostre script envia un salt de línia al final i, per tant,
    // totes les dades les podrem anar agafant de peticioAux
    // Si la darrera tasca no tingués salt de línia no ens arribaria
    // aquí però estaria guardada a peticio
    if ((ini_msg) && (comp_lin < mis_len) && (peticioAux.length() > 1)) {
        // Si la línia ja és la resposta la guardem
        missatge[comp_lin++] = peticioAux;
    }
    if (peticioAux.startsWith(F("HTTP/1.1 200"))) { // Resposta bona
        pendent = true;
        redir = false;
    }
    if (peticioAux.startsWith(F("HTTP/1.1 302"))) { // Redireccionament
        redir = true;
    }
    if (redir && (peticioAux.startsWith(F("Location:")))) {
        // Si hi ha redireccionament, hem de buscar l'adreça
        // i extreure'n el servidor i la pàgina
        String adre = peticioAux.substring(peticioAux.indexOf("/") + 2);
        server1 = adre.substring(0, adre.indexOf(".com") + 4);
        paginal = adre.substring(adre.indexOf(".com") + 4);
        server1.toCharArray(server, server_len);
        paginal.toCharArray(pagina, pag_len);
        ara = true;
    }
    if (pendent && peticioAux.startsWith(F("Connection: close"))) {
        ini_msg = true;
    }
    completa = false;
}
}
// Hi ha una resposta per processar
int contador = 1;
ocupat = false;
// Com mostrem la informació de la pantalla principal
if ((pendent) && (!pantallaSec)) {
    pendent = false;
    if (comp_lin > 0) {
        capTasca = false;
        // Si li info és la mateixa, no refresca pantalla
        for (byte k = 0; k < mis_len; k++) {
            if (missatgeAnterior[k] != missatge[k]) {
                tft.fillScreen(ILI9341_BLACK);
            }
            missatgeAnterior[k] = missatge[k];
        }
        for (byte k = 0; k < mis_len; k++) {
            if (missatge[k].length() > 1) { // Si no està buit
                Serial.print(F("--- Tasca "));
                Serial.print(k + 1);
                Serial.println(F(" ---"));
                processa(missatge[k]);
                contador += 1;
                altura_linia += 80;
                Serial.println();
            }
        }
    }
    if (peticio != "") {

```

```

        Serial.print(F("--- Tasca "));
        Serial.print(contador);
        Serial.println(F(" ---"));
        processa(peticio);
    }
    // Si la sala està lliure: cercle verd
    // Si la sala està ocupada: cercle vermell
    if (ocupat == false) {
        tft.fillCircle(280, 30 , 10, ILI9341_GREEN);
    } else {
        tft.fillCircle(280, 30 , 10, ILI9341_RED);
    }
} else {
    if (capTasca == false) {
        for (byte k = 0; k < mis_len; k++) {
            missatgeAnterior[k] = missatge[k];
        }
        tft.fillScreen(ILI9341_BLACK);
        tft.setCursor(40, 100);
        tft.setTextColor(ILI9341_WHITE);
        tft.print(F("Cap reserva pendent"));
        tft.fillCircle(280, 30 , 10, ILI9341_GREEN);
    }
    Serial.println(F("No queden tasques per al dia d'avui"));
    capTasca = true;
}
pensant = false;
}
// Com mostrem la informació de la pantalla secundària
if ((pendent) && (pantallaSec)) {
    pendent = false;
    if (comp_lin > 0) {
        capTasca = false;
        for (byte k = 0; k < mis_len; k++) {
            if (missatgeAnterior[k] != missatge[k]) {
                tft.fillScreen(ILI9341_BLACK);
            }
            missatgeAnterior[k] = missatge[k];
        }
        for (byte k = 0; k < mis_len; k++) {
            if (missatge[k].length() > 1) { // Si no està buit
                Serial.println(missatge[k]);
                processa(missatge[k]);
                pantallaSecOK = true; //Informació ben processada a la pantalla secundària
                altura_linia += 40;
                Serial.println();
            }
        }
    }
    if (peticio != "") {
        processa(peticio);
    }
    tft.drawRect(270, 200, 40, 30, ILI9341_BLUE);
    tft.setTextSize(1);
    tft.setCursor(276 , 210);
    tft.print(F("Torna"));
    tft.setTextSize(2);
}
pensant = false;
}

// Quan toca, tornem a fer una petició
if (ara || ((millis() - darreraConnexio > periodeConnexio))) {
    ini_msg = false;
    comp_lin = 0;
}

```

```

for (byte k = 0; k < mis_len; k++) {
    missatge[k] = "";
}
Serial.println(contTorna);
// Forma provisional fins que es canviï la freqüència de refresc
// Si contTorna no és parell canviarà quan està fent la connexió al servidor
// això provocaria un error ja que processaria la info amb pantallaSec=false
if (contTorna == 6) {
    pantallaSec = false;
    pantallaSecOK = false;
    contTorna = 0;
}
// S'ha programat un script de Google per cada pantalla
// Indiquem l'script corresponent a la pantalla principal i a la secundària
if ((!redir) && (!pantallaSec)) {
    server1 = "script.google.com";
    paginal = "/macros/s/AKfycbwNDGoOvY7UrZT_D-YNDYGxUFY4d79PxejRils4I74GzuvCJfU/exec";
    server1.toCharArray(server, server_len);
    paginal.toCharArray(pagina, pag_len);
}
if ((!redir) && (pantallaSec)) {
    server1 = "script.google.com";
    paginal = "/macros/s/AKfycbxENy5iznB1PX0fSXfWNDFUjHsSpqgwQ_fkZ3aJJmEcgrWDkZA/exec";
    server1.toCharArray(server, server_len);
    paginal.toCharArray(pagina, pag_len);
}
ara = false;
client.stop();
// Es fa la connexió al servidor per obtenir la informació dels calendaris
if (client.connect(server, 443)) {
    Serial.println(F("S'ha fet la connexió al servidor"));
    if (pantallaSec) {
        contTorna += 1;
    }
    pensant = true;
    client.print(F("GET "));
    client.print(pagina);
    client.println(F(" HTTP/1.1"));
    client.print(F("Host: "));
    client.println(server);
    client.println(F("Connection: close"));
    client.println();
    // Guardem quan hem fet la connexió
    darreraConnexio = millis();
} else {
    Serial.println(F("connection failed"));
    tft.fillScreen(ILI9341_BLACK);
    tft.setCursor(20, 20);
    tft.print(arreglaEsp("Connexió fallida"));
    // Si hi ha un error potser cal tornar a carregar el programa a l'Arduino
}

// Mirem si hi ha algun contacte amb el sensor tàctil
if (ts.bufferEmpty()) {
    return;
}
// Recuperem el punt
TS_Point p = ts.getPoint();
// Escalem les longituds de la pantalla per calibrar-la
// De ~0->4000 a longitud en pixels
p.x = map(p.x, TS_MINX, TS_MAXX, 0, tft.width());
p.y = map(p.y, TS_MINY, TS_MAXY, 0, tft.height());

```

```

// Definim la zona tàctil per passar de la pantalla principal a la secundària
if ((!pantallaSec) && (p.x > 230) && (p.y > 150) && (!pensant)) {
    tft.fillScreen(ILI9341_BLACK);
    tft.setCursor(50, 100);
    tft.setTextColor(ILI9341_WHITE);
    tft.print("Carregant aules...");
    pantallaSec = true;
    pensant = true;
}

// Definim la zona tàctil per passar de la pantalla secundària a la principal
if ((pantallaSecOK) && (p.x < 80) && (p.y > 140) && (!pensant)) {
    contTorna = 0;
    tft.fillScreen(ILI9341_BLACK);
    tft.setCursor(40, 100);
    tft.setTextColor(ILI9341_WHITE);
    tft.print("Carregant reserves...");
    pantallaSec = false;
    pantallaSecOK = false;
    pensant = true;
}

if (darrerEnces > millis()) { // Per si hi ha desbordament
    darrerEnces = millis();
}
if (digitalRead(sensorpir) == LOW) { // Sensor PIR detecta
    digitalWrite(BkLig, LOW); // Encén retroil·luminació
    darrerEnces = millis();
} else {
    if ((millis() - darrerEnces) > periodeEnces) { // Sensor PIR deixa de detectar
        digitalWrite(BkLig, HIGH); // Apaga retroil·luminació
    }
}
}
}

```

Annex 2: Script de Google: pantalla secundària

```

function doGet(e) {
    var numAules = 5;
    var nomsAules = new Array(numAules);
    nomsAules[0] = "Sala reunions";
    nomsAules[1] = "Aula 2.6";
    nomsAules[2] = "Aula 2.7";
    nomsAules[3] = "Aula LS.5";
    nomsAules[4] = "Aula LS.6";
    var resultat = "";
    var salt = "\n";
    var numCal = 99;
    var idCal = "";
    var ara = new Date();
    var Hora = ara.getHours();
    var Minut = ara.getMinutes();
    var HoraAra = Hora + "." + Minut;
    var nomCalendari = "";
    var Id = e.parameter.id;
    for (i = 0; i < numAules; i++){
        if (i == 0){ // Sala de reunions

```



```

        idCal = "p816s95q5215kvncilk0rbeao@group.calendar.google.com";
        numCal = 0;
        nomCalendari = "Sala de reunions";
    }
    if (i == 1){ // Aula 2.6
        idCal = "gmlg0vkthgqk6istln35r1a0uo@group.calendar.google.com";
        numCal = 1;
        nomCalendari = "Aula 2.6";
    }
    if (i == 2){ // Aula 2.7
        idCal = "v8ikfhlhv34o52imro6es4lips@group.calendar.google.com";
        numCal = 2;
        nomCalendari = "Aula 2.7";
    }
    if (i == 3){ // Aula LS.5
        idCal = "a6r6u9ndkokignltk59goiphqo@group.calendar.google.com";
        numCal = 3;
        nomCalendari = "Aula LS.5";
    }
    if (i == 4){ // Aula LS.6
        idCal = "225jjusk9vuh8mm6og6esk8e0g@group.calendar.google.com";
        numCal = 4;
        nomCalendari = "Aula LS.6";
    }
    if (numCal < numAules){
        var cal = CalendarApp.getCalendarById(idCal);
        var final = new Date();
        final.setHours(23);
        final.setMinutes(59);
        var esdev = cal.getEvents(ara, final);
        var numEsdev = esdev.length;
        if (numEsdev > 0){ // Hi ha, com a mínim, un esdeveniment
            if (esdev[0].getStartTime() < ara.getTime()) {
                resultat = resultat + nomCalendari + "," + dades(esdev[0]) + "," + "0";
                // 0 vol dir "En curs"
            } else {
                resultat = resultat + nomCalendari + "," + dades(esdev[0]) + "," + "1";
                // 1 vol dir "Lliure"
            }
            resultat = resultat + salt;
        } else {
            resultat = resultat + nomCalendari + "," + " " + "," + " " + "," + "2" + salt;
            // 2 vol dir "Lliure tot el dia"
        }
    }
}
return ContentService.createTextOutput(resultat);
}

```

```
function dades(esdAct) { // Organitza les dades d'un esdeveniment
    var dataIni = esdAct.getStartTime(); // Data i hora d'inici
    var dataFi = esdAct.getEndTime(); // Data i hora d'acabament
    var DosDigitsIni = dataIni.getMinutes().toLocaleTimeString().replace(/:/g, ''); // Adaptació de l'hora al format HH:MM
    var DosDigitsFi = dataFi.getMinutes().toLocaleTimeString().replace(/:/g, '');
    var ini = dataIni.getHours() + "." + DosDigitsIni + dataIni.getMinutes();
    var fi = dataFi.getHours() + "." + DosDigitsFi + dataFi.getMinutes();
    var resul = ini + "," + fi;
    return resul;
}
```

Annex 3: Script de Google: pantalla principal

```
var idCal = "3gmV3l0bfvm4kqbcnhd0atfo@group.calendar.google.com";
// Script per interactuar amb el calendari
// Funció que s'executa quan hi ha una ordre get
function doGet(e) {
    var salt = "\n";
    var cal = CalendarApp.getCalendarById(idCal);
    if (!cal) { // Si el calendari no existeix o no tenim permís
        resultat = "Calendari no trobat!";
        return ContentService.createTextOutput(resultat);
    }
    var ara = new Date(); // La data i l'hora del moment d'executar l'script
    var final = new Date();
    final.setHours(23); // Li canviem l'hora a les 23.59
    final.setMinutes(59);
    // Agafem tots els esdeveniments des d'ara (inclosos els ja iniciats) fins les 23.59 h
    var esdev = cal.getEvents(ara, final);
    var numEsdev = esdev.length; // Quants n'hi ha?
    var resultat = "";
    if (numEsdev > 0) { // Hi ha, com a mínim, un esdeveniment
        resultat = resultat + dades(esdev[0]); // Primer esdeveniment
        if (esdev[0].getStartTime() < ara.getTime()) {
            resultat = resultat + ", " + "0"; // 0 vol dir "En curs"
        }
        resultat = resultat + salt;
    }
    if (numEsdev > 1) { // Hi ha, com a mínim, un segon esdeveniment
        resultat = resultat + dades(esdev[1]) + salt; // Segon esdeveniment
    }
    if (numEsdev > 2) { // Hi ha, com a mínim, un tercer esdeveniment
        resultat = resultat + dades(esdev[2]); // Tercer esdeveniment
    }
    return ContentService.createTextOutput(resultat); // Enviem la resposta
}
function dades(esdAct) { // Organitza les dades d'un esdeveniment
    var descrip = esdAct.getTitle(); // Títol de l'esdeveniment
    if (descrip == '') {
        var descrip = 'Reserva sense nom'
    }
    var dataIni = esdAct.getStartTime(); // Data i hora d'inici
    var dataFi = esdAct.getEndTime(); // Data i hora d'acabament
    var DosDigitsIni = dataIni.getMinutes().toLocaleTimeString().replace(/:/g, ''); // Adaptació de l'hora al format HH:MM
    var DosDigitsFi = dataFi.getMinutes().toLocaleTimeString().replace(/:/g, '');
    var ini = dataIni.getHours() + "." + DosDigitsIni + dataIni.getMinutes();
    var fi = dataFi.getHours() + "." + DosDigitsFi + dataFi.getMinutes();
    var resul = descrip + ", " + ini + ", " + fi;
    return resul;
}
```

}